Individual Coursework 5DATA004W Data Science Project Lifecycle

**Name** : Kelvin Rushbrook

**Student ID** : W18984891

**Dataset** : Meteorite Landings <https://catalog.data.gov/dataset/meteorite-landings>

*Please note , it seems that since starting this coursework , the website now throws a “404 not found” error when trying to download the file.*

*This only started occurring toward the end of the coursework and is out of my control, however you can still download the dataset from the GitHub repository at location* [*https://github.com/Kelstudy/MeteorDashboard/blob/main/Meteorite\_Landings.csv*](https://github.com/Kelstudy/MeteorDashboard/blob/main/Meteorite_Landings.csv)

**Link to video presentation :**

ADD

**Link to GitHub Repository :**

<https://github.com/Kelstudy/MeteorDashboard/tree/main>

**Link to Streamlit app:**

<https://meteordashboard-6gr5xubacyqqshvgbpuivp.streamlit.app/>

# Aims and Objectives:

Meteorites provide valuable insights into the origins, composition, and history of the solar system. Public datasets, such as the “Meteorite Landings” dataset from data.gov, enable researchers to study trends and anomalies without relying solely on fieldwork.

This project aims to create an interactive dashboard that not only brings these insights to the general public but also serves as a valuable tool for the scientific community.

### Aims:

* Design and interactive dashboard that allows exploration of historical meteorite landings, thus helping users identify patterns within the data
* Encourage public interest in astronomy by making meteorite data visually engaging and intuitively interactive.
* Provide policy makers with clear data that can assist in education.
* Identify outliers and trends in the dataset

### Objectives:

* Allow users to filter the data and visualisations, by discovery date, mass , and meteorite class by using intuitive sidebar controls
* Provide an interactive global map using Pydeck , to plot meteorite landings based on geolocation data
* Present key statistics via KPIS
* Present trends via bar charts , line charts , and summary tables
* Summarize key statistics and findings within an “Key insights” page, offering interpretations of the data.

### Justifications for key insights:

The “Key insights" page was designed to highlight meaningful discoveries that could guide a better scientific understanding and allow guided policy planning.

All of these insights were selected not only for statistical, practical interest but also to keep the user engaged and deepen the general publics understanding of meteorites.

Insights include :

**Heaviest Meteorite (Hoba)**:This was used as it helps to highlight a significant outlier in the dataset , which dramatically influences the average mass.

Including this , helps to display just how large meteorites can get, and also helps to show users the extreme impact an outlier like this , can have on the average.

**Average Mass of Meteorites:** Showcasing the average mass gives users a general idea of the typical size of meteorites. This helps correct misconceptions that all meteorites are extremely large, by showing that most are relatively small compared to famous examples. This can be used in conjunction with the Heaviest meteorite insight , to allow users to see just how large the mass differences can be.

**Oldest Recorded Meteorite (Nogata, year 860):** Identifying the oldest recorded meteorite landing shows the long history of meteorite observations, allowing the viewer to see the lasting human interest in astronomy.

**Most Common Meteorite Class (L6):** This insight helps to show the predominance of a specific type of meteorite material reaching Earth. This insight informs scientists about the typical composition of meteorite falls and can guide further studies.

**Total Number of Meteorites and Unique Classes:** Presenting the total number of recorded meteorites alongside the number of unique classes shows the diversity in the dataset. It helps scientists to analyse just how varied meteorites can be.

**Percentage of Meteorites with Location Data:** Showing the percentage of meteorites that have geographic coordinates , helps underline the dataset’s quality and completeness. It allows users to trust the visualizations , knowing that the majority of the meteorites within the dataset , have been located exactly.

### Development Methodology

For this project, I used an agile-inspired solo method.

I chose to use this method as it allows for flexibility and is iterative, meaning I can focus on working software instead of documentation.

This is ideal while working solo within a timeframe , when the main priority is a working dashboard.

The key elements of Agile that I applied where:

* **Iterative development :** I decided to split my project into small code blocks called regions, this allowed me to focus on getting specific sections of the project done and fully completed before moving to the next stages. For example , creating the sidebar was its own region , as was creating the key insights page.
* **Rapid Prototyping :** The immediately updating nature of Streamlit , meant I could work with rapid prototyping , making a change in the code ,meant I could immediately visualise this on the Streamlit app and see how it looked , if it worked correctly etc.
* **Flexibility for change:** Building on top of the rapid prototyping and code block usage , this also allowed for a lot of flexibility in changing my code while still maintaining other sections of my code and keeping them tidy.

GitHub acted as the central version control repository.

With each new key implementation or change, my code was updated within GitHub.

This worked well as GitHub directly linked to Streamlit Cloud , so with each push, Streamlit Cloud updated and allowed me to test all was working seamlessly.

### Requirements:

Goal : Develop a dashboard that lets users easily view , filter and analyse meteorite landing data , to support education and research.

**Functional requirements:**

These are features that the application must deliver in order to meet the goal.

1. Users must be able to filter meteorite data by year ranging , using a slide bar
2. Users must be able to filter meteorite data by mass range , using numeric input
3. Users must be able to filter meteorite data by a single meteorite class, or all classes, by using text input for selecting a single class, or a tick box for all classes.
4. Users must be able to view , zoom , and interact with a PyDeck map which displays meteorite landing locations with tooltips.
5. Key statistics are displayed using KPIS.
6. User must be able to navigate between 4 pages , “Overview” , “Charts” , “Data Tables” , and “Key Insights”.
7. Visualisations such as charts , maps and KPIs must dynamically change to reflect the current filter

**Non-Functional requirements:**

These are features that define how the system should behave and perform, including aspects like performance and reliability.

1. The application must load in under 10 seconds for the standard user
2. The layout should be intuitive and require no prior training for efficient use.
3. All graphs , maps and KPIs should be responsive and update within 5 seconds
4. Data filtering must maintain data accuracy
5. App performance must remain stable with datasets up to 50,000 rows
6. Code should be well documented

### Test Cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | TC1 | Title: | | Year filter test |
| Description | | | Ensure the year range filter updates all displayed data correctly | |
| Steps and input data | | | 1. Select a specific year range using the sidebar year filter 2. Observe updates in map , KPIs , charts and data tables | |
| Dependencies | | | The “Year” field in the dataset , must exist and be numeric data type | |
| Expected results | | | map , KPIs , charts and data tables should update with correct year range once filter is applied | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | TC2 | Title: | | Mass filter test |
| Description | | | Ensure the mass range filter updates all displayed data correctly | |
| Steps and input data | | | 1. Select a specific mass range using the sidebar mass filter 2. Observe updates in map , KPIs , charts and data tables | |
| Dependencies | | | 1. The “Mass” field in the dataset , must exist and be numeric data type 2. The “Mass (g)” field in the dataset must be converted to be called (mass\_g) in the code , to prevent issues with spaces in the column name | |
| Expected results | | | Map , KPIs , charts and data tables should update with correct mass range once filter is applied | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | TC3 | Title: | | Class filter test |
| Description | | | Ensure the class range filter updates all displayed data correctly | |
| Steps and input data | | | 1. Select a meteorite class using dropdown 2. Observe updates in map , KPIs , charts and data tables | |
| Dependencies | | | The “recclass” field in the dataset , must exist and be populated | |
| Expected results | | | Map , KPIs , charts and data tables should update with correct class once filter is applied | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | TC4 | Title: | | Map functionality test |
| Description | | | Ensure the PyDeck map renders correctly , shows meteorite location data , and display tooltips | |
| Steps and input data | | | 1. Navigate to overview page 2. Zoom in and out on map 3. Hover over a meteorite point to ensure tooltip displays | |
| Dependencies | | | Reclat and reclong must exist as fields in the dataset and be populated with numeric data | |
| Expected results | | | Map renders with meteorite markers and tooltips on hover | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | TC5 | Title: | | Key insight static data test |
| Description | | | Ensure the Key insights data does not change based on sidebar filters | |
| Steps and input data | | | 1. Navigate to key insights page 2. Change filters to random in sidebar 3. Ensure data does not change on key insights page | |
| Dependencies | | | Original\_df dataset must remain unfiltered | |
| Expected results | | | Key Insight data remains unchanged regardless of filters. | |

### Test Log:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC | Date | Executed by | Actual Result | Pass/Fail | Notes |
| TC1 | 28/04/2025 | Kelvin Rushbrook | All visuals updated correctly based on year | Pass | NA |
| TC2 | 28/04/2025 | Kelvin Rushbrook | All data fell within filtered mass range | Pass | NA |
| TC3 | 28/04/2025 | Kelvin Rushbrook | Only selected class displayed | Pass | NA |
| TC4 | 28/04/2025 | Kelvin Rushbrook | Map loaded and properly and tooltips showed | Pass | NA |
| TC5 | 28/04/2025 | Kelvin Rushbrook | Insights were static and not affected by filters | Pass | NA |